

Modifikácia kódu procesu počas behu

Samuel Čavoj
školiteľ Ing. Dušan Bernát, PhD.

Cieľ

- Vyvinúť čo najvšeobecnejšiu sadu nástrojov
- Opravovanie chýb v kóde počas behu
- Zmena správania procesu, pridávanie novej funkcionality

Existujúce riešenia

- Prevažne v priestore jadra (kpatch, Ksplice)
- Katana

Bakatsugi ばかつぎ

- Nástroj injektuje kód z knižnice do cieľového procesu
- Berie PID a knižnicu s metadátami (patch object)
- Nahrádzanie knižničných funkcií a vlastných funkcií procesu

Predpoklady

- Linux na platforme x86_64
- Administrátorské privilégia
- Dynamicky linkované ELF procesy (GNU libc)
 - Prístup k tabuľke symbolov

Prostriedky na Linuxe

Niektoré použité prostriedky:

- `ptrace(2)`
 - Zastavenie procesu a prvotná manipulácia stavu
 - Injektovanie prvej fázy (stage 1)
- `unix(7)`
 - Komunikácia medzi injektorom a cieľom
 - Posielanie otvorených súborov (file descriptor)
- `memfd`
 - Súbory nachádzajúce sa v pamäti
- `dlopen(3)`
 - Načítanie knižnice počas behu
- `proc(5)`
 - Získavanie informácií o ciele
 - Prístup k otvoreným súborom a spustiteľnému súboru cieľa

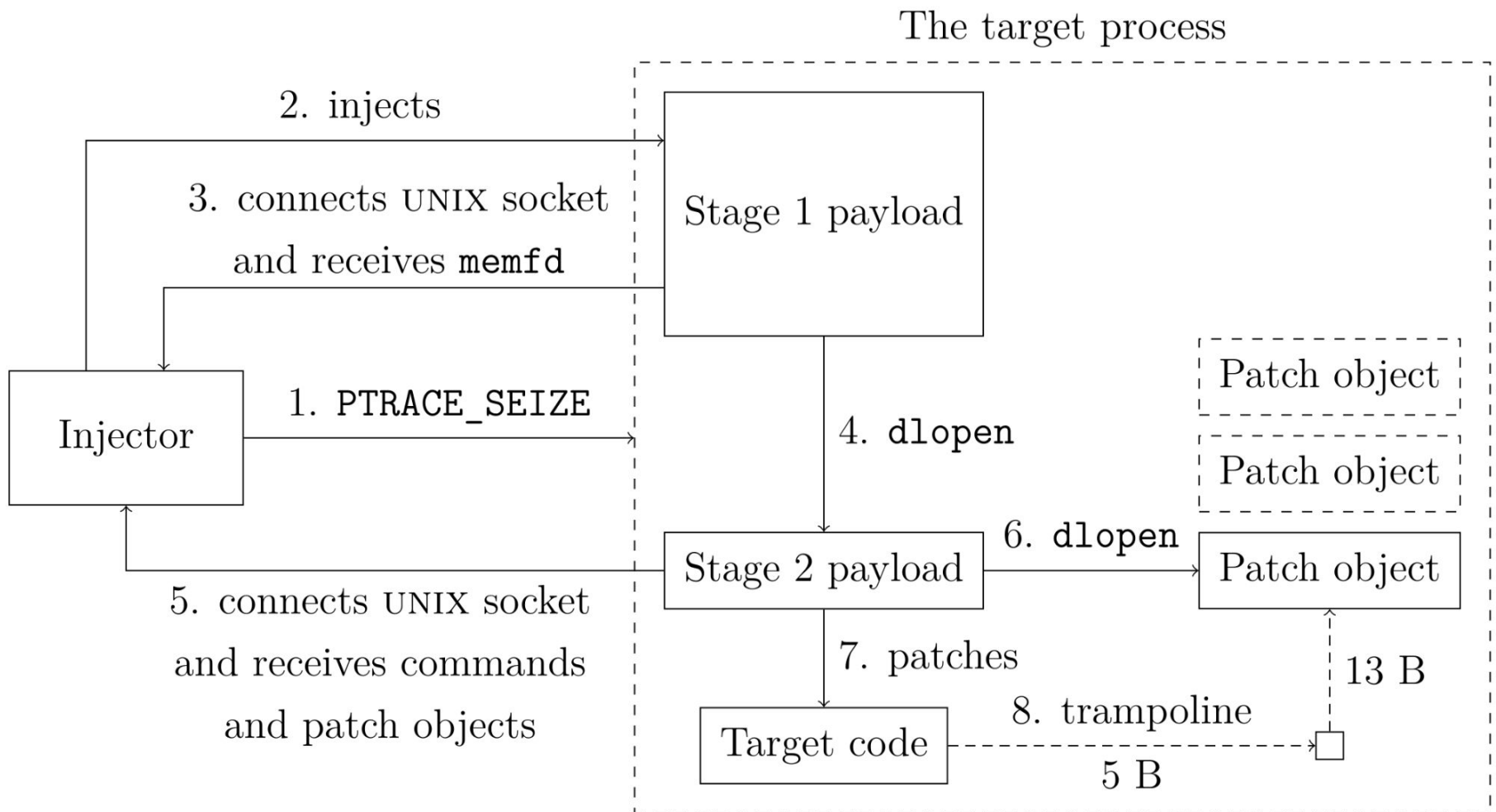
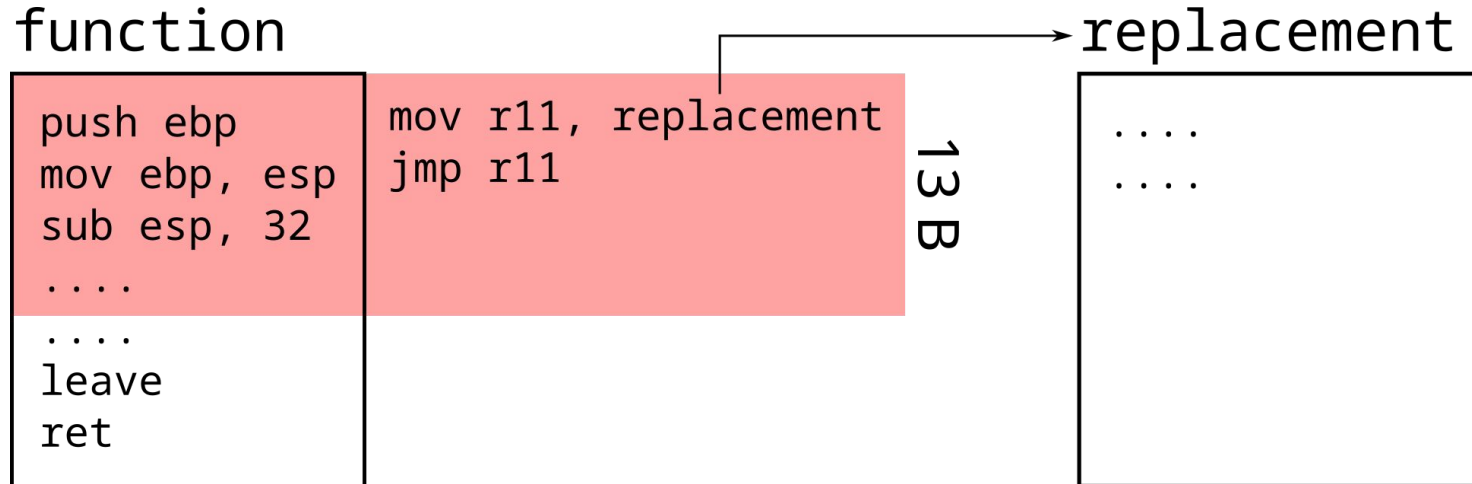


Diagram popisujúci architektúru a fungovanie bakatsugi

Nahrádzanie vlastných funkcií

Úprava volaní je nerealistická, bránia aj nepriame volania.

Prepíšeme začiatok funkcie trampolínou, ktorá skočí na novú implementáciu.



Nahrádzanie vlastných funkcií

Menšia trampolína (5 B)

function

```
push ebp
mov ebp, esp
sub esp, 32
....
....
leave
ret
```

```
jmp rel trampoline
```

5 B

trampoline

```
mov r11, replacement
jmp r11
```

13 B

replacement

```
....
....
```

Ostatné nástroje

- C hlavička (.h) pre tvorbu záplatových knižníc (patch object)
 - Kompilácia ako pri bežnej zdieľanej knižnici
 - Metadáta pomocou makier z hlavičky
- Nástroj `bakals` na zobrazenie symbolov

```
#include <sys/time.h>
#include "bakatsugi.h"

int r_gettimeofday(struct timeval *restrict tp, void *restrict tzp) {
    int r = gettimeofday(tp, tzp);
    tp->tv_sec -= 759;
    return r;
}

BAKATSUGI(
    PATCH_LIB(gettimeofday, r_gettimeofday)
)
```

```
$ gcc -o libpatch.so libpatch.c -fPIC -shared -I[path-to-header]
```


Príklad

```
int x(int a) {
    return a + 1;
}

int main() {
    int i = 0;
    while (1) {
        printf("%d\n", x(i++));
        sleep(1);
    }
}
```

```
#include "bakatsugi.h"
```

```
int x(int a) {
    return a * 2;
}
```

```
BAKATSUGI(
    PATCH_OWN(x, x)
)
```

```
$ ./example
```

```
1
```

```
2
```

```
3
```

```
stack_mapping: MapRange { range_start:
140722665209856, range_end: 140722665345024,
offset: 0, dev: "00:00", flags: "rw-p", inode:
0, pathname: Some("[stack]") }
rsp: 0x7ffc8c7a99c0
Checking at 0x7ffc8c7aa000
Checking at 0x7ffc8c7ab000
PayloadData { cookie: [50, 82, 76, 53, 108,
103, 53, 76, 88, 118, 66, 107, 86, 65, 114,
71], was_syscall: true }
Got fd: 5
Opening patch lib: /proc/self/fd/5
target at: 55a811d24149
replacement at: 7f584e86d0e9
trampoline allocator base: 0x55a811d22000
trampoline allocator mmap: 0x55a811d22000
rip: 0x55a811d2414e, addr: 0x55a811d22000
distance: -8526
trampoline: [e9, b2, de, ff, ff]
```

```
6
```

```
8
```

```
...
```